REMARKS

Applicant respectfully requests the reconsideration of this application and the consideration of the following remarks.

Specification

The Office Action requested the applicant to "update the status of all copending application made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate." Applicant is not aware of any particular instance in the application that needs an appropriate update. If the examiner is actually aware of any instance that needs an update, Applicant respectfully requests the examiner to particularly point out the instance. If the examiner is not actually aware of any instance that needs an update, Applicant respectfully requests the withdrawal of the objection to the specification.

Double Patenting Rejections

Claims 14, 16, 18, 23, 26, 27, 33 were provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1, 4, 5, 1, 4, 5 and 11, respectively, of copending application no. 10/038,478. Applicant respectfully disagrees.

Claims 1, 4, 5 and 11 of copending application no. 10/038,478 claim various methods, which are statutory ***process*** claims. Claims 14, 16, 18, 23, 26, 27 and 33 of the present application claim various execution units, which are statutory ***product*** claims. A process cannot be the same as a product. Thus, claims 14, 16, 18, 23, 26, 27 and 33 of the present application and claims 1, 4, 5 and 11 of copending application no. 10/038,478 are clearly directed to different subject matters. It is clearly improper to assert that they are the "same invention". Thus, at least for these reasons, the withdrawal of the rejection based on double patenting of the "same invention" type is respectfully requested.

The Office Action asserted that claims 18, 19 and 27 are substantial duplicate of claims 16, 17 and 26. Applicant respectfully disagrees.

Claims 16 and 18 recite:

16. (original) An execution unit in a microprocessor, the execution unit comprising:
a plurality of look-up tables;
a first circuit coupled to the plurality of look-up tables and a Direct Memory Access (DMA) controller, the first circuit, in response to the microprocessor receiving a single instruction, replacing at least one entry <u>in at least one of</u> the plurality of look-up tables with <u>at least one data element</u> using the DMA controller.

18. (original) An execution unit in a microprocessor, the execution unit comprising:
a plurality of look-up tables;
a first circuit coupled to the plurality of look-up tables and a Direct Memory Access (DMA) controller, the first circuit, in response to the microprocessor receiving a single instruction, replacing at least one entry <u>for each of</u> the plurality of look-up tables with <u>a plurality of data elements</u> using the DMA controller.

Applicant respectfully submits that "***at least one of***..." is clearly different in scope from "***each of***...", and "***at least one*** data element" different in scope from "***a plurality of*** data elements". Thus, it is erroneous to assert that claim 18 is a substantial duplicate of claim 16.

Claims 17 and 19 depend from claims 16 and 18 respectively to incorporate the limitations their parent claims. Thus, claims 17 and 19 are different in scope as claims 16 and 18 are different in scope.

Claims 26 and 27 recite:

26. (original) An execution unit in a microprocessor, the execution unit comprising:
means for replacing at least one entry in <u>at least one of</u> a plurality of look-up units in a microprocessor unit with <u>at least one number</u> using a Direct Memory Access (DMA) controller;
wherein the above means operate in response to the microprocessor receiving a single instruction.

27. (original) An execution unit in a microprocessor, the execution unit comprising:
means for replacing at least one entry for <u>each of</u> a plurality of look-up units in a microprocessor with <u>a plurality of numbers</u> using a Direct Memory Access (DMA) controller;
wherein the above means operate in response to the microprocessor receiving a single instruction.

Applicant respectfully submits that "***at least one of*** ..." is clearly different in scope from "***each of*** ...", and "***at least one*** number" different in scope from "***a plurality of*** numbers". Thus, it is erroneous to assert that claim 27 is a substantial duplicate of claim 26.

<u>35 U.S.C. §112 Rejections</u>

Claim 73 was rejected for reciting "... whether any bit after the last bit of input is used in obtaining the first result".

The Office Action asserted that "A last bit is the final bit and there are no more bits after it" (see, e.g., items 11 and 12 of the Office Action). Applicant respectfully submits that such an interpretation is improper.

Applicant respectfully requests the reconsideration of claim 73 in view of claim 72, the limitation of which is incorporated into claim 73 through claim dependency. Claims 72 and 73 recite:

> 72. (previously presented) An execution unit as in claim 47 further comprising:
> means for receiving a first number indicating <u>a position of a last bit of input in the string of bits</u>.
>
> 73. (original) An execution unit as in <u>claim 72</u> further comprising:
> means for generating an indicator indicating whether <u>any bit after the last bit of input</u> is used in obtaining the first result.

In view of what is recited in claim 72 (e.g., a position of ***a last bit of input*** <u>in the string of bits</u>), the meaning of "any bit after ***the last bit of input***" is clear. The interpretation of "A last bit is the final bit and there are no more bits after it" for the claim limitation is erroneous and inconsistent with what is explicitly recited in the claim.

During patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000) (See also, e.g., MPEP 2111). The meaning of the claim limitations should be construed in a way consistent with the specification.

For example, the specification shows "The number of bits between bit 8073 and the last bit in entry vA is FenceBits, which is stored in bit segment 8022 of entry vC" (page 103, lines 11-12) and "Bit 8073 represents the last valid bit for code words in the bit stream" (page 103, lines 8-9).

Thus, the "FenceBits" indicates the position of a last bit (8073) of input in the string of bits provided in the register entry vA. In general, the last bit (8073) of the bitstream (8009) is not necessarily the last bit of the string of bits provided in the register entry vC. There could be more bits after the last bit (8073) in register entry vA, as clearly illustrated in Figure 60.

The description of "FenceBits" (8022) of Figure 60, "Fence Bits" (7901) of Figure 59, etc., clearly conveys to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention and enables one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Thus, the withdrawal of the rejection under 35 U.S.C. 112 is respectfully requested.

35 U.S.C. §103(a) Rejections

Claims 16, 18, 26-32 were rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,397,324 (hereinafter "Barry") in view of U.S. Patent No. 5,768,628 (hereinafter "Priem"). Applicant respectfully disagrees.

Note that claims 17 and 19 depend from claims 16 and 18 respectively. Claims 17 and 19 were rejected under 35 U.S.C. 102(e), while claims 16 and 18 are not anticipated by Barry, as admitted in Office Action. Since claims 17 and 19 incorporate the limitation of claim 16 and 18 through claim dependency, it is clear that the rejections under 35 U.S.C. 102(e) for claims 17 and 19 are improper.

The examiner bears the initial burden of ***factually*** supporting any prima facie conclusion of obviousness under 35 U.S.C. 103. A prima facie case of obviousness is established by presenting evidence that ***would have led*** one of ordinary skill in the art to combine the relevant teachings of the references to arrive at the claimed invention. **It is impermissible to simply make a hindsight reconstruction of the claimed invention using the claim as a template and filling the gaps using the elements from the references.**

"The tendency to resort to hindsight upon applicant's disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art." (MPEP 2142).

MPEP (2141) shows that "When applying 35 U.S.C. 103, the following tenets of patent law must be adhered to: (A) The claimed invention must be considered as a whole; (B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination; (C) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and (D) Reasonable expectation of success is the standard with which obviousness is determined."

Further, MPEP (2142, 2143) shows that "To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure."

A broad conclusory statement regarding the obviousness of modifying a reference, standing along, is not "evidence".

The S2TBL and STBL instructions of Barry copy data from registers into the processor's local memory. A person skilled in the art would not use DMA in such operations of Barry, which is indicated by the lack of a description of the modification as suggested in the Office Action. Thus, the modification suggested in the Office Action was clearly a hindsight reconstruction aimed only at filling the gaps using the elements from the references.

Further, the disclosure of Barry shows that Barry et al. were aware of the use of DMA. For example, Barry explicitly shows the DMA (see, e.g., Figure 1, DMA 181 of Barry) in the system of Barry. It is apparent that Barry et al. understood the use of DMA. Barry explicitly mentioned DMA but designed the system in a way different from what is claimed. Thus, the fact that Barry failed to describe the subject matter as claimed provides the clear indication of non-obviousness.

Based on the description of Priem and Barry, it does not appear that the description of Priem would inspire Barry et al. further toward the subject matter as claimed, since Barry et al. were aware of the use of DMA.

Further, Barry was filed in 2000 which was many years after Priem was filed in 1995 and issued in 1998. Applicant respectfully request the examiner consider the fact that Barry actually describes the used of a DMA but fail to present the subject matter as claimed. Priem was filed in 1995 and issued in 1998, many years before Barry was filed in 2000. Barry et al. would be considered as persons skilled in the art. However, Barry et al. failed to design an execution unit as recited in the claims 16-19 and 26-32, even though Barry et al. had the knowledge about DMA (see, e.g., Figure 1, DMA 181) in general. This fact is a clear indication of non-obviousness.

Thus, applicant respectfully submits that the modification suggested in the Office Action was based on the hindsight afforded by the teaching of the present invention and the specific arrangements as recited in claims 16-19 and 26-32 were not obvious in view of Barry and Priem. Applicant respectfully submits that 16-19 and 26-32 are patentable over Barry and Priem.

35 U.S.C. §102(e) Rejections

Claims 1-15, 20-25, 33-35, 37-41, 43-56 and 61-85 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,397,324 (hereinafter "Barry"). Applicant respectfully disagrees.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). (See, e.g., MPEP 2131)

Further, during patent examination, the pending claims must be "given their broadest reasonable interpretation consistent with the specification." In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000). (See, e.g., MPEP 2111)

Applicant respectfully submits that Barry does not show each and every aspect of any of claims 1-15, 20-25, 33-35, 37-41, 43-56 and 61-85. Applicant respectfully requests the ***reasonable*** interpretation ***consistent*** with the specification.

For example, claim 1 recites:

1. (original) An execution unit in a microprocessor, the execution unit comprising:
look-up memory; and
a first circuit coupled to the look-up memory,
the first circuit, in response to the microprocessor receiving a first instruction, partitioning the look-up memory into a first plurality of look-up tables,

the first circuit, in response to the microprocessor receiving a second instruction, partitioning the look-up memory into a second plurality of look-up tables which are different from the first plurality of look-up tables.

In the specification, for example, Figures 38 and 39 illustrate examples of circuits that actually configure look-up units into different look-up tables. The example circuits include MUXes (e.g., 5851, 5831, ..., 5831, 5832, ... in Figure 38; 5931, 5941, ... in Figure 39), address select (e.g., 5871 in Figure 37; 5971 in Figure 38) and output select (e.g., 5873 in Figure 37; 5973 in Figure 38), etc.

Claim 1 recites "the first circuit, in response to the microprocessor receiving a first instruction, partitioning the look-up memory ***into a first plurality of look-up tables***" and "the first circuit, in response to the microprocessor receiving a second instruction, partitioning the look-up memory ***into a second plurality of look-up tables*** which are different from the first plurality of look-up tables".

For example, in the circuit of Figure 39, in one configuration, "8 look-up units LUT0 - LUT7 are used as 8 independent look-up tables" (Paragraph [00252], the specification). In another configuration specified by an instruction, "look-up units LUT0 - LUT7 are used as 2 independent look-up tables, each of which contains 512 16-bit entries" (Paragraph [00253], the specification). Further, "if a vector register has 128 bits, two units as shown in **Figure 39** may be used to simultaneously look up a vector of 16 8-bit elements from 16 256-entry tables in one configuration; a vector of 4 16-bit elements from 4 512-entry tables in another configuration; and a vector of 2 16-bit elements from 2 1024-entry tables in another configuration" (Paragraph [00254], the specification).

In Figures 38 and 39 of the present application, one can clearly see examples of circuits that partition the set of look-up units into different sets of look-up tables. However, Barry shows only memory banks that are designed to be addressed independently from each other. In Barry, there is no similar circuitry as illustrated in the present applicant and recited in the claim.

In Barry, for example in Figure 1, the local memory is connected to the memory interface through fixed signal lines. Assuming that the banks of local memory of Barry corresponding to the limitation of "look-up memory", the rejection is at most based on ***speculation***, which is improper, when failing to point to a specific ***circuit*** of Barry that corresponds to "a first circuit coupled to the look-up memory, the first circuit, in response to the microprocessor receiving a first instruction, partitioning the look-up memory into a first plurality of look-up tables, the first circuit, in response to the microprocessor receiving a second instruction, partitioning the look-up memory into a second plurality of look-up tables which are different from the first plurality of look-up tables".

Further, for example, claim 4 recites:

> 4. (previously presented) An execution unit as in claim 1 wherein the look-up memory comprises <u>a plurality of look-up units</u>, and wherein <u>the first circuit is to configure the plurality of look-up units into a third plurality of look-up tables</u> in response to the microprocessor receiving a third instruction.

However, in Figure 1 of Barry, no specific circuit can be identified as corresponding to "the first circuit" that is to "configure the plurality of look-up units into a third plurality of look-up tables". Barry does not show the specific structural arrangement as recited in claim 1. In Barry, if the memory banks were considered as look-up units, there is no ***circuit*** that configures the memory banks into different sets of look up tables. The instructions of Barry being logically described as accessing different sets of look up tables, in terms of logic functions, are not evidence of particular arrangements of circuits.

Further, there is evidence that Barry does not have the first circuit as recites in the claim. For example, Col. 11, lines 33-38, of Barry shows

> "Quad Address Table Look-Up (L4TBL) To support this instruction type, the SP and each PE data memories are split into four separate banks which are addressable independently. The addressing mechanism is organized in a similar manner to the dual table apparatus with extensions to support four banks of memory bank-0 to bank-3." (Col. 11, lines 33-38, Barry)

From this description, it is understood that the SP and each PE data memories of Barry are ***split***, by design, into four separate banks, bank-0 to bank-3, in order to support the L4TBL instruction.

Thus, according to the description of Barry (e.g., Col. 11, lines 9-10), bank-0 and bank-1 are used to support the L2TBL instruction. If the processor by design has only two independently addressable banks, bank-0 and bank-1, the L4TBL instruction is not supported. To support the L4TBL instruction according to Barry, the processor is to be designed to have four independently addressable banks, bank-0 to bank-3.

Thus, it is clear that, to design the processor according to Barry, a human designer arranges to split the data memories into separate banks that are independently addressable. The processor of Barry cannot and does not split the memories into separate, independently addressable banks in response to either the L2TBL instruction or the L4TBL instruction.

Therefore, from the description of Barry (Col., 10, lines 24-45; Col. 10, line 62 to Col. 11, lines 48; and Col. 11, line 65 to Col. 12, line 27) about the instructions (e.g, L4TBL), separating memory into separate banks is a design choice, which is a decision made by a human but not an operation performed by the processor in response to receiving an instruction. Barry does not show ***the first circuit*** "partitioning look-up memory into a plurality of look-up tables" "in response to the microprocessor receiving the single instruction".

Thus, at least for the above reasons, the withdrawal of the rejection under 35 U.S.C. 102(e) for claim 1 is respectfully requested.

Further, for example, claim 2 recites:

> 2. (original) An execution unit as in claim 1 wherein <u>a total number of bits used by each entry in the first plurality of look-up tables is different from a total number of bits used by each entry in the second plurality of look-up tables</u>; and wherein the microprocessor is a media processor formed in a monolithic semiconductor substrate, <u>which comprises a memory controller for controlling host memory</u>, said media processor being coupled to said memory controller.

The Office Action asserted that "a L2TBL and a L4TBL instruction can use different number of bits for each entry." However, such an assertion, even if correct, is about instructions. A person skilled in the art understands that instructions are not circuits. Instructions may be implemented in different ways. Barry does not specifically show a circuit that partitions look up memory into look up tables of different entry size in a way illustrated in the Figures of the present application and recited in claim 2.

Rejections based on speculations are not proper. The description about the instructions in Barry is clearly insufficient for the anticipation of claim 2. Barry does not show "a first circuit" which partitions memory into look-up tables of different entry sizes, in response to different instruction. Thus, Barry does not have an execution unit as claimed in claim 2.

Further, memory banks 431/433 of Figure 4 of Barry are processors' local memory, not host memory; and memory interface unit 485 of Figure 4 of Barry is not a memory controller for controlling host memory.

The Office Action asserted that "Host memory is just memory that is used independently". The Office Action pointed to "Dictionary of computers, information processing, and telecommunications" for the definition of "host", which shows:

> "Host (H): an information processor which performs the instruction processing work of the enterprise. A host processor is generally self-sufficient and requires no supervision from other processors. See *data host*. See also *host computer, host interface, host node, host processor, host system*"

In view of the above definition of "host", it appears that the Examiner improperly extrapolated the meaning from "host processor" into "host memory". A person skilled in the art understands that processors' local memory is clearly not "host memory". Applicant respectfully requests ***reasonable*** interpretation ***consistent*** with the specification. The Office Action asserted that "Host memory is just memory that is used independently" without specifying from what it is used independently. While it may make sense to say a host process can operate independently from other processors. It makes no sense to define "host memory" as a memory that can be used independently from other memories. The Office Action's position appears to be that "host memory is just memory", which is clearly unreasonable.

Applicant respectfully submits that the processors' local memory as in Barry is clearly not "host memory".

Thus, at least for the above reasons, claim 2 is patentable over Barry.

Further, for example, claim 3, recites:

3. (original) An execution unit as in claim 1 wherein a total number of entries in each of the first plurality of look-up tables is different from a total number of entries in each of the second plurality of look-up tables.

Barry does not have a circuit that can reconfigure the memory banks into look-up tables of different total entry numbers according to instructions.

Col. 11, lines 13-14, of Barry shows "Maximum architecture defined size of the LUT is 64 K entries." Col. 11, lines 44, of Barry shows "Maximum size of the LUT for this case is 256 entries." Clearly, these are architectural limitations on the possible sizes of logical LUTs that can be implemented in Barry. Once a human designer completes the design, the sizes are determine and cannot be changed according to the instructions.

Note that in claim 3, the different sizes of the LUTs are the results of the first circuit partitioning the look-up memory differently. In Barry, the architectural limitations are the results of design choice of a human being. Thus, the elements of Barry used for the rejection do not correspond properly to the claim limitations. The rejection is clearly improper.

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987) (See, e.g., MPEP 2131).

Applicant respectfully requests the explicit and consistent identification of what in Barry correspond to those recited in the pending claims, since it appears that the Office Action still frequently applied the elements of Barry in an inconsistent way, even though the previous response of the Applicant had already pointed out a number of inconsistent points.

One example of an inconsistent way is to consider limitation X as corresponding to element A of Barry in reading a portion of a claim and then switch to a different viewpoint of considering the same limitation X as corresponding to element B of Barry in reading another portion of the same claim. The inconsistency indicates that either the elements of Barry cannot be applied as proposed since at least one of the elements does not correspond to the limitations properly, or the applied elements do not satisfy each and every aspects of the claim since certain relations as recited in the claim are missing/broken. Thus, such an inconsistent way of applying elements of a reference is clearly improper for making a rejection.

For example, item 116 of the Office Action considered "an entry of a register file" of claim 78 as corresponding to an instruction register; and the instruction register is considered as a single register register file. Since claim 79 is a dependent claim of claim 78, the claim limitations of claim 78 are incorporated into claim 79 through claim dependency. Thus, for a proper rejection, "the entry of the register file" in claim 79 should be consistently considered as the instruction register that is the single register register file. However, claim 79 recites:

> 79. (original) An execution unit as in claim 78 wherein the single instruction specifies an index of the entry in the register file.

Since the instruction register is a single register register file, there appears no need for an instruction to specify an index of the single entry. Since it is a single register register file, the index would be the index for the instruction itself. Since it is a single register register file, the index would be always the same. It makes no sense for an instruction to specify its own location in the single register register file (the instruction register). Further, Barry does not show an instruction that specifies an index of the single entry in the "single register register file", the instruction register.

In item 117, in the context of claim 79, the Office Action asserted that "The Rz register provides the index not the sign extension bit". It appears that the Office Action was switching from considering "the entry in the register file" (e.g., limitation X) as the single entry in the "single register register file" (e.g., element A of Barry) to considering "the entry in the register file" (e.g., the same limitation X) as the Rz register (e.g., element B of Barry).

Further inconsistent points in the Office Action are discussed below. Applicant respectfully requests explicit and consistent identification of what in Barry correspond to those recited in the pending claim.

Claim 20 recites:

> 20. (previously presented) An execution unit in a microprocessor comprising:
> a plurality of look-up tables;
> a first circuit coupled to the plurality of look-up tables, the first circuit configured to <u>receive a string of bits</u>;
> a second circuit coupled to the plurality of look-up tables and the first circuit, the second circuit configured to receive a plurality of data elements, in response to the microprocessor receiving a single instruction,
> the second circuit generating a plurality of indices using the plurality of data elements and the string of bits,
> the plurality of look-up tables looking up simultaneously a plurality of entries using the plurality of indices; and
> a third circuit coupled to the plurality of look-up tables, the third circuit combining the plurality of entries into a first result.

Applicant respectfully submits that it is improper to consider the data in Rze and Rzo as both the "string of bits" and "the plurality of data elements" (see, e.g., item 119 of the Office Action). The Office Action (item 119) asserts that "The Rz register is partitioned in to a plurality of segments that are the indices. The indices are data elements ...". Thus, the Office

Action considered the data in Rze and Rzo as corresponding to three claim limitations: "a plurality of indices", "a plurality of data elements" and "a string of bits".

However, claim 20 recites "the second circuit generating a plurality of indices ***using*** the plurality of data elements ***and*** the string of bits". Note that claim 20 recites the word "***and***", not "or". Thus, "the plurality of data elements" and "the string of bits" are different from each other. It is improper to consider a same element of the reference as corresponding to two different elements of the claim. It is improper to consider the same data in Rze and Rzo as corresponding to different data recited in the claim.

For example, in Figure 60 of the present application, instruction vvld for the variable length decoding of a bit stream (8009) uses local control (8030) which includes L0-L7 (8031-8038). "Each local control information Lx (8015) contains: i) IgnBits (8054) indicating the number of bits to be ignored after the bit pointed to by Ptr (8047); and ii) IdxBits (8055) indicating the number of bits of the bit segment to be used in the construction of the index for one of the look-up tables." (Paragraph [00283], the specification)

Thus, in the example of Figure 60, it is clear that the plurality of local control information L0-L7 (8031-8038) is different from the bit stream (8009).

Further, for example, claim 87 recites:

> 87. (new) An execution unit as in claim 20 wherein a first one of the plurality of indices is generated from retrieving a first bit segment from the string of bits according to a first one of the plurality of data elements.

Barry does not show "retrieving a first bit segment from the string of bits according to a first one of the plurality of data elements".

Further, for example, claim 88 recites:

88. (new) An execution unit as in claim 87 wherein a second one of the plurality of indices is generated from retrieving a second bit segment from the string of bits according to a second one of the plurality of data elements; and the first bit segment of the string of bits and the second bit segment of the string of bits have different bit lengths.

In Barry, indices in Rze and Rzo are of an equal bit length. Thus, Barry does not have "the first bit segment of the string of bits and the second bit segment of the string of bits have different bit lengths".

Further, for example, claim 89 recites:

89. (new) An execution unit as in claim 20 wherein the string comprises a plurality of variable length codes; and the plurality of data elements specify bit lengths of the variable length codes.

In Barry, indices in Rze and Rzo are of an equal bit length. Thus, Barry does not have a string that has a plurality of variable length codes, where "a plurality of data elements specify bit lengths of the variable length codes". Clearly, the indices in Rze and Rzo of Barry do not specify lengths of variable length codes. Note that claim 20, the parent claim of claim 89, recites: "the second circuit generating a plurality of indices using the plurality of data elements and the string of bits". It is clearly improper to consider that the indices in Rze and Rzo correspond to both the string that "comprises a plurality of variable length codes" and the plurality of data elements that "specify bit lengths of the variable length codes".

Further, for example, claim 48 recites:

48. (previously presented) An execution unit as in claim 47 further comprising:
means for receiving a plurality of data elements specifying the plurality of segments in the string of bits.

In Barry, the indices in Rze and Rzo are exactly the bit segments themselves.

Similarly, for example, claim 90 recites:

> 90. (new) An execution unit as in claim 47 further comprising:
> means for receiving a plurality of data elements specifying <u>how the string of bits is segmented for the plurality of segments respectively</u>.

In Barry, the indices in Rze and Rzo do not specify how the Rz is segmented into Rze and Rzo respectively. In Barry, how Rz is segmented into Rze and Rzo is pre-designed.

Further, for example, claim 86 recites:

> 86. (new) An execution unit as in claim 20 wherein the third circuit <u>combines</u> the plurality of entries into the first result <u>before a result based on the first result is outputted into an entry of a register file</u>.

For example, the Office Action considered that storing the looked up results into register Rt as corresponding to "combining the plurality of entries into a first result". Clearly, storing the looked up results into register Rt cannot be considered as corresponding to "the third circuit combines the plurality of entries into the first result ***before*** <u>a result based on the first result is outputted into an entry of a register file</u>".

Further, for example, claim 21 recites:

> 21. (original) An execution unit as in claim 20 further comprising:
> a fifth circuit coupled to the second circuit, the fifth circuit configured to receive at least one format; and
> a sixth circuit coupled to the fifth circuit and the third circuit, in response to the microprocessor receiving the single instruction,

<u>the fifth circuit formatting the string of bits into at least one escape data using the at least one format</u>, and
<u>the sixth circuit combining the at least one escape data with the first result into a second result</u>.

In item 35.b, the Office Action asserted that "The H1 portion is then combined with the H0 portion to make a sign-extended result in Rz (Barry column 11, lines 15-27)". However, in Barry (col. 11, lines 15-27), there is no indication that the result is stored in Rz. In Barry the results of the L2TBL instruction are stored in Rt, not in Rz. Thus, this mention of "Rz" is considered as a typographical error in the present response. Multiple inconsistent points in the Office Action regarding claim 21 are discussed below.

From the description of Barry (col. 11, lines 15-27), the sign-extension operation is performed on the data looked up from the memory banks, which are clearly not data loaded from Rz (Rze and Rzo). In rejecting claim 20, the Office Action considered that the string of bits corresponding to the data in Rz (Rze and Rzo). Since claim 21 is a dependent claim of claim 20, and thus incorporating the limitations of claim 20 through claim dependency, the Office Action should consistently considered the string of bits as the data in Rz (Rze and Rzo). However, it is apparently that the Office Action, in item 35.b, switched viewpoints from considering the string of bits as the data in Rz (Rze and Rzo) to considering the string of bits as data looked up from the memory bank. Such inconsistent viewpoints are clearly improper, as discussed above.

Further, as pointed out in the previous response, the sign-extension bit in Barry is specified for all of the look up results for the instruction. Thus, if "the plurality of entries" stored into the register Rt were considered as "the first result" and the sign extended entities as "the at least one escape data", there is no combination of "the plurality of entries" with "the sign extended entities" in Barry. The results of the instruction of Barry are either entirely sign-extended or entirely non-sign-extended.

Applicant respectfully requests the consistent and explicit identification of what in Barry correspond to those recited in the pending claim, such as "string of bits", "first result", "second result", etc. The Office Action failed to recognize the inconsistency in the rejections, after the previous response had pointed out the inconsistency. The Office Action merely reiterated what can be found in Barry, ignored the relations between parts of the claim as if different parts of the claims were separate and independent from each other. When the elements of Barry cannot be correlated with the claim limitations in a way as recited in the claim, the rejection is improper. Thus, applicant respectfully requests consistent and explicit identification of what in Barry correspond to those recited in the pending claim.

The above discussion for claims 20 and 21 also applies to claims 47 and 76, claims 9-12.

Further, for example, claim 51 recites:

> 51. (original) An execution unit as in claim 48 further comprising:
> means for receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements.

The Office Action (item 120) took the position that "The indices are pointers". There are a number of inconsistent points in such a position.

For example, in Barry, the indices are not "***bit*** pointers". An index of Barry is not used to point to a bit in a bit stream. At most, the indices of Barry might be considered as "look up table entry pointers". A person skilled in the art understands that an entry pointer is not a bit pointer. Not that claim 51 recites "a ***bit*** pointer", not just "pointer".

Further, for example, claim 51 specifies that "the plurality of segments in the string of bits are determined *using the bit pointer **and** the plurality of data elements*". The Office Action took the position that the indices in Rz (Rze and Rzo) correspond to "the plurality of segments in the string of bits", "the bit pointer" and "the plurality of data elements". In such a position, what is to be determined is exactly the same as each of two used for the determination. It is not clearly how such a position could justify the correspondence with the limitation of "***using** … **and** …*".

Further, for example, claim 52 recites:

52. (original) An execution unit as in claim 51 further comprising:
means for generating a new bit pointer <u>using the first result</u>.

The Office Action considered the results in the register Rt as corresponding to the first result; and the indices in Rz as corresponding to the bit pointer. However, claim 52 recites "means for <u>generating</u> *a new bit pointer* <u>using the first result</u>." The Office Action did not show what is Barry corresponds to "a new bit pointer" generated using the results in the register Rt.

Further, claim 63 recites:

63. (original) An execution unit as in claim 47 wherein the means for combining the plurality of entries comprises:
means for selecting a valid data from the plurality of entries.

The Office Action asserted

> "When the data is retrieved from the data table, it is selecting data from the table, since it is choosing one, two, or four entries out of the all the entries in the tables, and the data is always seen as valid. There is nothing in Barry to indicate that that data is never considered valid" (item 121, the Office Action).

There are multiple inconsistent points in this position. The above-recited position of the Office Action in reading the additional limitation recited in claim 63 is not consistent with position of the Office Action in reading the claim limitations incorporated from claim 47.

Applicant respectfully submits that claim 47 recites "means for looking up simultaneously <u>a plurality of entries</u> from a plurality of look-up tables using the plurality of indices". Claim 63 depends from claim 47 to incorporate the limitations of claim 47 through claim dependency. Thus, the plurality of entries are the results looked up from the look-up tables.

Apparently, in considering claim 63 the Office Action switched into the position of considering "all the entries in the tables" as corresponding to the limitation of "the plurality of entries", which is not consistent with what is recited in claim 47, which specifies "means for looking up simultaneously <u>a plurality of entries</u> from a plurality of look-up tables using the plurality of indices". In claim 63, the term "the plurality of entries" refers to "<u>a plurality of entries" introduced in claim 47.</u>

Claim 47 further recites "<u>the means for combining</u> the plurality of entries <u>comprises</u>" "means for selecting a valid data from the plurality of entries". Since the Office Action considered outputting the entries into register Rt as corresponding to "combining the plurality of entries", it is clear that "choosing one, two, or four entries out of the all the entries in the tables" is not "selecting a valid data" from the results looked up from the look-up tables.

Furthermore, "choosing one, two, or four entries out of the all the entries in the tables" is the same operation as "looking up simultaneously <u>a plurality of entries</u> from a plurality of look-up tables" but not a part of outputting the entries into the register Rt. Thus, the above-recited position of the Office Action introduces a further inconsistent point, since the Office Action considered outputting the entries into the register Rt as corresponding to "combining the plurality of entries comprises" and claim 47 requires "<u>the means for</u>

combining the plurality of entries ***comprises***" "means for selecting a valid data from the plurality of entries".

Thus, it is clear that the Office Action failed to consistently apply the elements of Barry in a number of ways in rejection claim 63.

Further, claim 72 recites:

> 72. (previously presented) An execution unit as in claim 47 further comprising:
> means for receiving a first number indicating a position of a last bit of input in the string of bits.

In the present application, an example of the last bit of input is "Bit 8073" that "represents the last valid bit for code words in the bit stream" (page 103, lines 8-9; see also Figure 60). In Figure 60 of the present application, it is seen that the last bit of input in the string of bits is not necessarily the last bit in register vA.

The Office Action took the position that the size field indicates the position of the last bit, which is improper. Such "last bit" as considered in the Office Action is not "a last bit of input in the string of bits", because the Office Action took the position that the data in register Rz (Rze and Rzo) correspond to the string of bits. In Barry, the last bit of input in register Rz is always the last bit of the register Rz. Thus, the Office Action failed to apply the elements of Barry in a consistent way for the rejection.

Further, claim 73 recites:

> 73. (original) An execution unit as in claim 72 further comprising:
> means for generating an indicator indicating whether any bit after the last bit of input is used in obtaining the first result.

The Office Action rejected claim 73 under 35 U.S.C. 102 as being anticipated by Barry. However, the Office Action failed to specify what in Barry correspond to the additional limitations recited in claim 73. Thus, the rejection is improper.

The Office Action took the position of "a last bit is the final bit and there are no more bits after it" in making the rejection of claim 73 under 35 U.S.C. 112. As discussed above, such an interpretation is improper, since claim 72 recites "means for receiving a first number indicating a position of a last bit of input in the string of bits". See, e.g., the description about "FenceBits" (8022) and "Fence Bits" (7901).

Since the additional limitation recited in claim 73 is supported by the description of the present application but not in the Barry, the withdrawal of the rejections for claim 73 is respectfully requested.

Further, claims 74-75 recite:

74. (previously presented) An execution unit as in claim 47 further comprising:
means for generating an indicator indicating whether one of <u>the plurality of segments of bits</u> contains a predetermined code.

75. (original) An execution unit as in claim 74 wherein the predetermined code represents an end of block condition.

The Office Action asserted:

"Barry states in column 11, lines 13-14 that the maximum architecture size of the look-up table is 64 k entries. This means that the indices cannot surpass the 64 K entry limit, i.e., there is an end limit to the indices, which is an end of block condition."

Applicant respectfully submits that it appears the Office Action rejected claim based on matching only one limitation "an end of block condition" to the "64 K entry limit" of Barry, while ignoring other limitations. Such a position is clearly improper for an anticipation-based rejection, since for a proper anticipation-based rejection the single reference must show each and every element as set forth in the claim.

For example, claim 75 is a dependent claim of claim 74, the limitation of which is incorporated into claim 75 through claim dependency. Thus, claim 75 requires: "means for generating an indicator indicating whether <u>one of the plurality of segments of bits</u> contains a predetermined code", "wherein the predetermined code represents an end of block condition". However, from the above-recited position of the Office Action, there is no indication of "an indicator indicating whether one of <u>the plurality of segments of bits</u> contains a predetermined code" that represents an end of block condition.

In the Office Action, the data in register Rz (Rze and Rzo) are considered as the plurality of segments of bits. However, from the description of Barry and the above-recited position of the Office Action, it is not clearly what actually corresponds to "a predetermined code" recited in the claim. Assuming that the number "64 K" were considered as corresponding to the "predetermined code", it is clear that the number represents the end of the table, since it is the architectural limit.

Further, in Barry, there is no description that shows the generation of an indicator to indicate whether one of the data in Rze and Rzo contains the number "64 K". Speculation is not evidence; and speculation cannot be used for rejection.

In general, Applicant respectfully submits that Barry, there is no description that shows the generation of an indicator to indicate whether one of the data in Rze and Rzo contains a predetermined code. Thus, Barry does not have "means for generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code"

Furthermore, as pointed out above, the number "64 K" would represent the end of the look up table. However, a person skilled in the art understands that an "end-of-block" code is a particular, predetermined code in a stream of codes (e.g., a variable length code stream) to represent the end of block condition. For example, a bit stream may include multiple blocks of variable length codes; and the end of each of the blocks is marked by the particular, predetermined code (EOB) that represents the end of block condition.

Thus, applicant respectfully submits that, in the Office Action, the interpretation for the term "end of block" is incorrect.

The Office Action pointed to Col. 15, lines 13-17 and Col. 15, line 63 – Col. 16, line 4 of Barry for the rejection of claims 74-75. However, this description of Barry is substantially irrelevant to claims 74-75, since claims 74-75 recite "the plurality of segments of bits", which corresponds to "a plurality of segments of bits in the string of bits" recited in claim 47. Consider that the indices in Rze and Rzo were considered as corresponding to "the plurality of segments of bits", it is clear that Barry does not have, in an execution unit, "means for generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code", such as a code for an end of block condition.

For example, claim 33 recites:

33. (original) An execution unit in a microprocessor, the execution unit comprising:

means for receiving a plurality of numbers;

<u>means for partitioning look-up memory into a plurality of look-up tables;</u>

means for looking up simultaneously a plurality of elements from the plurality of look-up tables, each of the plurality of elements being in one of the plurality of look-up tables and being pointed to by one of the plurality of numbers;

<u>wherein the above means operate in response to the microprocessor receiving a single instruction</u>.

The Office Action relied on Barry (Col. 7, lines 54-62 and Col. 9, lines 63-67) for the limitation of "partitioning look-up memory into a plurality of look-up tables". However, the description of Col. 7, lines 54-62 and Col. 9 lines 63-67 of Barry does not correspond to the claim limitation, since claim 11 recites the limitation "the above operations are performed in response to the microprocessor receiving the single instruction", which applies to "partitioning look-up memory into a plurality of look-up tables".

The Office Action relied on the L2TBL instruction of Barry for the rejection of claim 11. However, the description of Col. 7, lines 54-62 and Col. 9, lines 63-67 of Barry does not show "partitioning" in response to the L2TBL instruction. Col. 7, lines 54-62, of Barry shows the hardware design choice of using "a multiple bank memory that makes it possible to generate multiple independent data-dependent load and store operations". Col. 9, lines 63-67, of Barry shows the design choices of using "both memory bank-1 431 and memory bank-0 433 simultaneously to support two load operations in parallel or two store operations in parallel". These design choices of using a memory with multiple banks cannot be considered corresponding to "*partitioning* look-up memory into a plurality of look-up tables, ... wherein the above operations are performed ***in response to the microprocessor receiving the single instruction***".

Furthermore, a designer is not an execution unit. The designer cannot be considered as corresponding to "means for partitioning ..." in an execution unit of a microprocessor.

Note that, in response to the L2TBL instruction, the processor of Barry uses the base addresses stored in the address register file (ARF) and offsets stored in the computer register file (CRF) to access the multiple bank memory. Since the look-up tables in the multiple bank memory of Barry are imaginary, based on how one interprets the meaning of the base addresses and offsets, the processor of Barry does not perform "partitioning look-up memory into a plurality of look-up tables, ... wherein the above operations are performed in response to the microprocessor receiving the single instruction".

Claim 14 and 23 recite the limitation of "wherein the microprocessor is a media processor integrated with ***a memory controller for host memory*** on a single integrated circuit". The memory interface unit (485) of a Barry is not a memory controller for host memory. The memory interface unit (485) is for local memory banks (431 and 433). Barry does not show "a media processor integrated with a memory controller for host memory on a single integrated circuit".

Further, claim 80 recites:

> 80. (previously presented) An execution unit as in claim 33 wherein the look-up memory comprises a plurality of look-up units, and wherein the means for partitioning look-up memory comprises:
> <u>means for configuring the plurality of look-up units into the plurality of look-up tables</u>.

In Barry, the independently addressable memory banks might be considered as corresponding to the plurality of look-up units recited in the claim. However, in Barry, the connection between the local memory interface unit and the memory banks are fixed signal lines. There is no "means for configuring the plurality of look-up units into the plurality of look-up tables" in Barry. The description of accessing multiple logical look-up tables through accessing memory banks via a generic memory controller is not sufficient to anticipate the claim limitation the circuit. Note that speculations are improper for rejections.

Further, as discussed above, Barry uses memory bank-0 and bank-1 for L2TBL and memory bank-0 to bank-3 for L4TBL. Separation of the local memory into bank-0 to bank-3 is a design choice. Barry does not show "configuring the plurality of look-up units into the plurality of look-up tables".

Other claims recite the limitations discussed above or indirectly contain the above discussed limitations through dependency to the above discussed limitations. Thus, the pending claims are patentable over Barry.

Please charge any shortages or credit any overages to Deposit Account No. 02-2666. Furthermore, if an extension is required, Applicant hereby requests such extension.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 6/9, 2005

[illegible]

Lehua Wang
Reg. No. 48,023

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-1026
(408) 720-8300